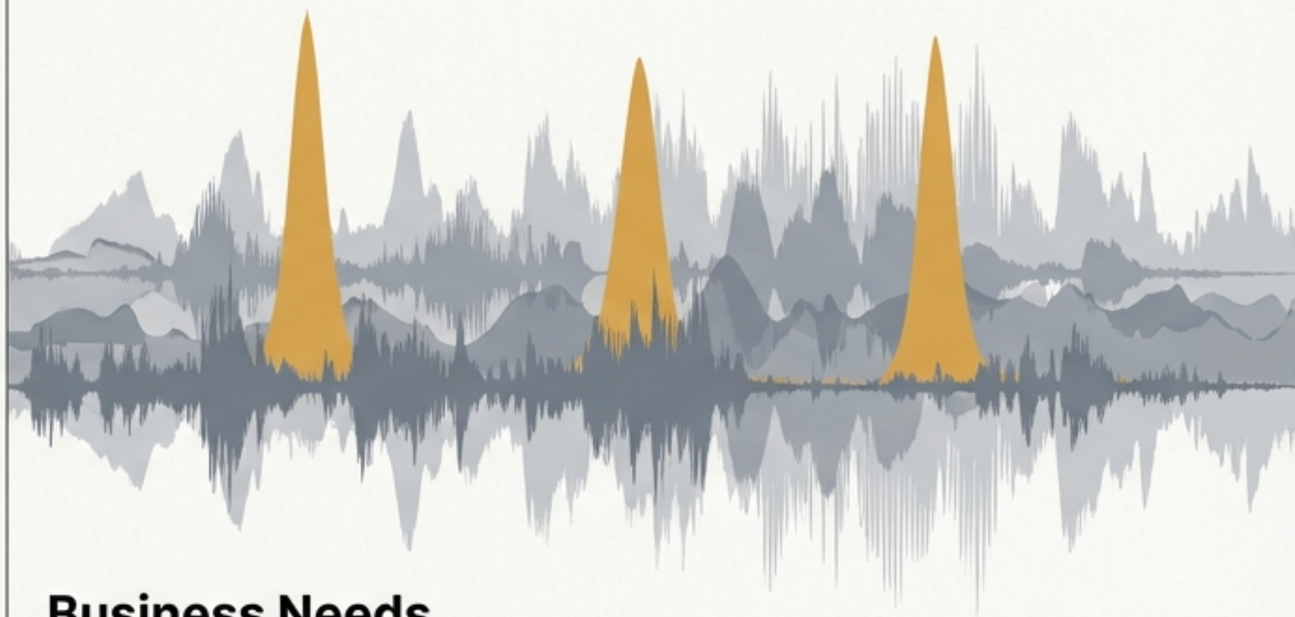
End-to-End Anomaly Detection & AI-Powered Explanation in Snowflake

A technical walkthrough of a call center solution built with Snowpark, Cortex, and `mistral-large2`.

The Supervisor’s Challenge: Finding and Understanding Outliers at Scale

Core Problem Description

Call centers generate massive volumes of interaction data. Within this data, critical outliers—such as calls with extremely long handle times, unusual knowledge management (KM) usage, or repeated escalations—are difficult to systematically identify and diagnose.



Business Needs

- ✓ **Automated Detection:** A scalable method to flag anomalous calls without manual review.
- ✓ **Root Cause Understanding:** A clear way to understand *why* a call was flagged as anomalous.
- ✓ **Actionable Insights:** Guidance on what actions to take, from agent coaching to process improvement.

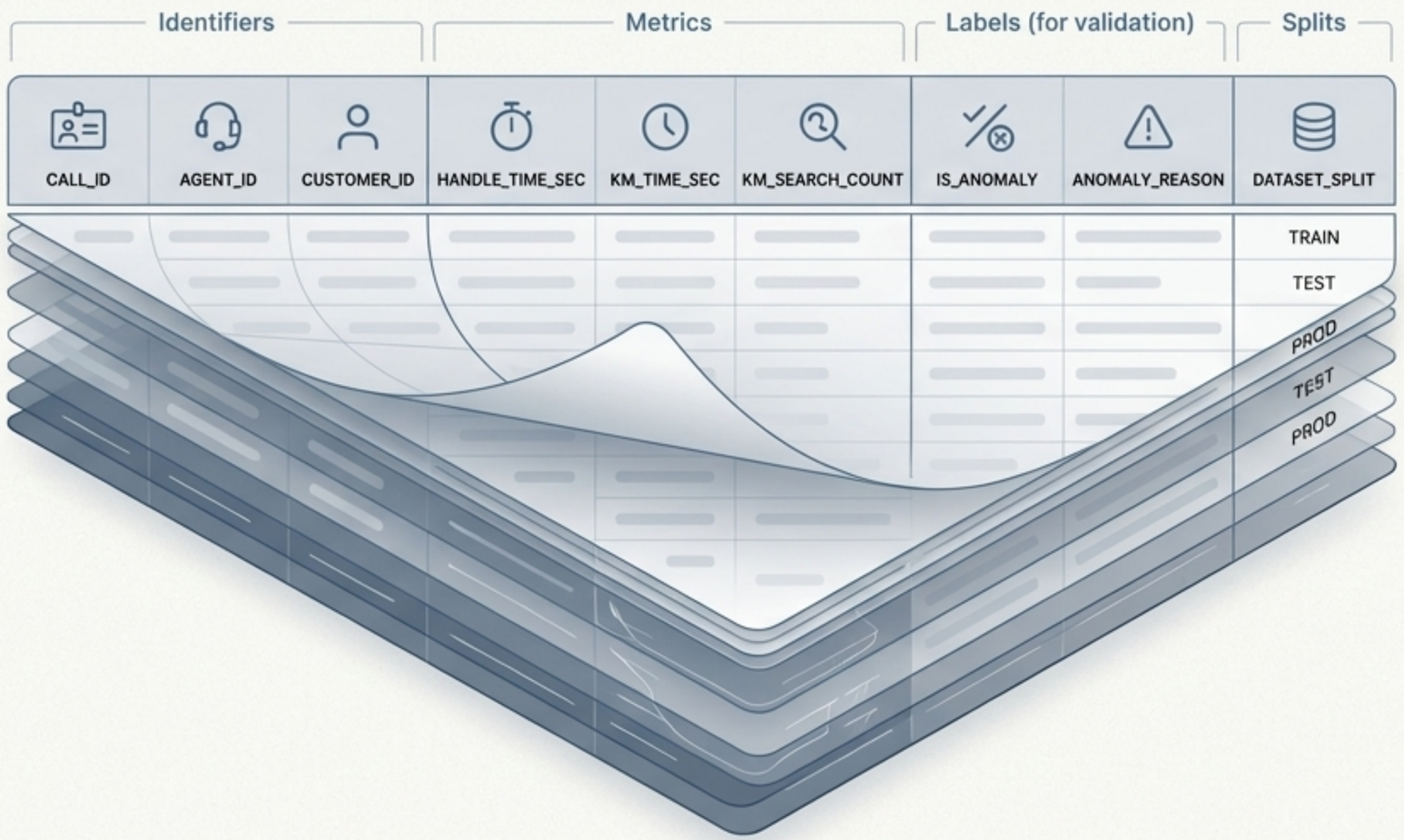
Building the Foundation: A Synthetic Call Center Dataset

Rationale

To develop and validate our models without using real customer data, we engineered a synthetic dataset simulating 12 months of activity (~100,000 call records).

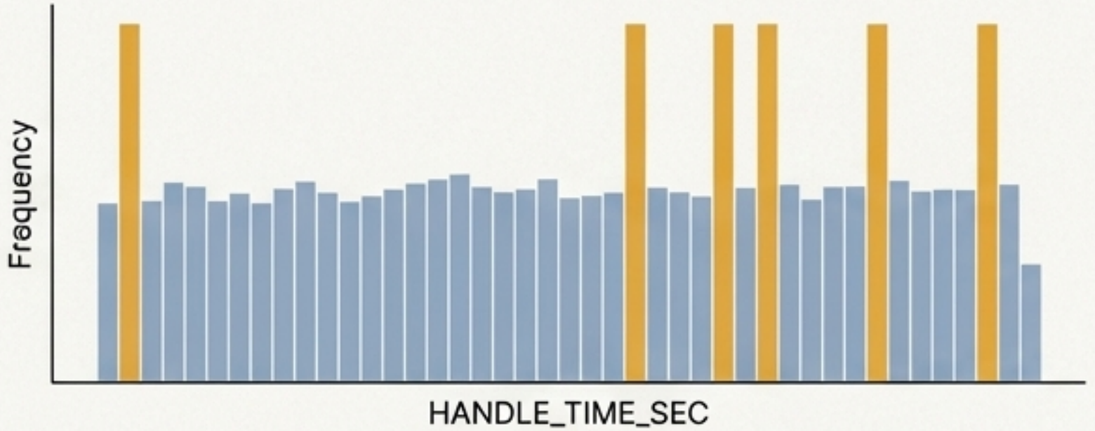
Core Table

`LAB_CALL_CENTER_KM`



Engineering a Labeled Dataset for Model Development

Anomaly Injection Strategy



We intentionally label ~10% of calls as anomalous with `ANOMALY_REASON = 'ExtremeLongHandleTime'`.

These records have massively inflated `HANDLE_TIME_SEC` values compared to the 'Normal' population.

This creates a clean, labeled ground-truth set for evaluating model performance.

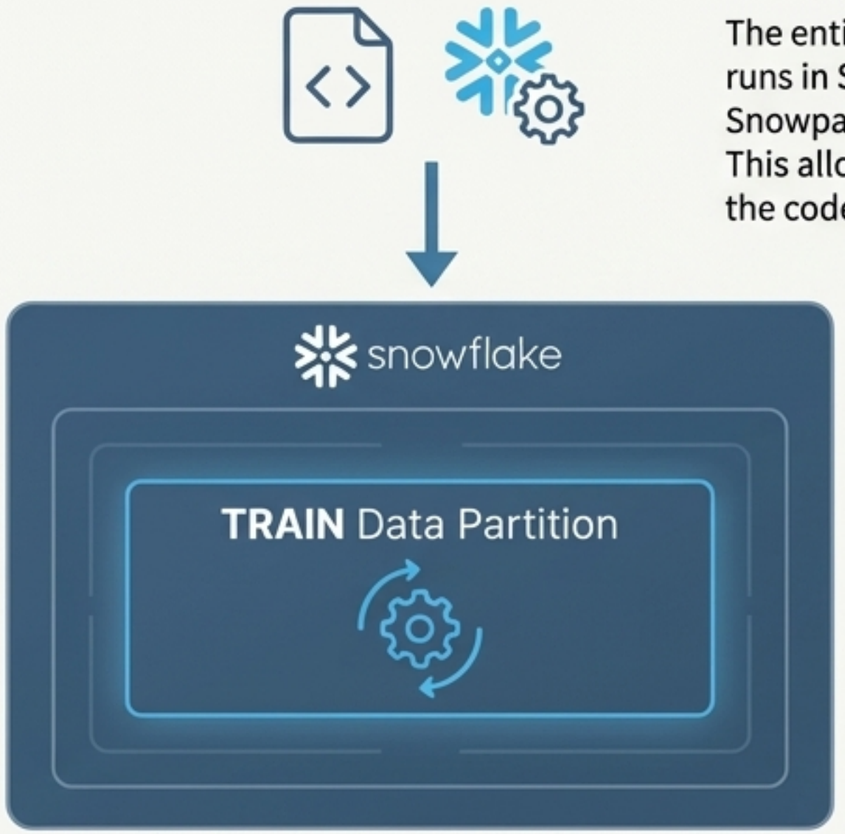
Time-Based Data Splitting

The `DATASET_SPLIT` column partitions the data chronologically using `CALL_DATE`.

This structure supports realistic model training, validation, and simulated production scoring.

Training the Detection Model with Snowpark and Scikit-learn

Code to Data Flow



The entire ML workflow runs in Snowflake using Snowpark for Python. This allows us to bring the code to the data.

Section 1: Feature Selection

We use a numeric feature set derived from `LAB_CALL_CENTER_KM`, including:

- `HANDLE_TIME_SEC`
- `KM_TIME_SEC`
- `AFTER_CALL_WORK_SEC`
- `HOLD_TIME_SEC`
- `KM_SEARCH_COUNT`
- `AGENT_TENURE_MONTHS`
- `HOUR_OF_DAY`

Section 2: Algorithm

We use an **Isolation Forest**, an unsupervised algorithm well-suited for anomaly detection.

- **Key Parameter:** `contamination=0.10`, set to align with our known 10% anomaly injection rate.

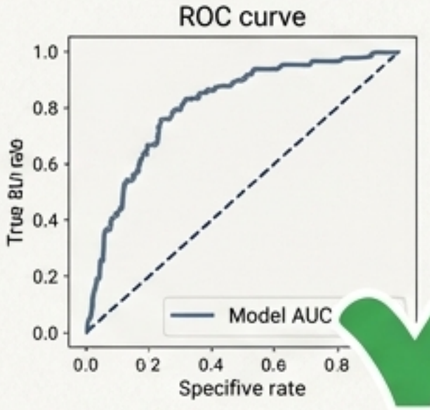
The model is fit exclusively on the `TRAIN` data partition.

Evaluating Performance and Scoring Production Data

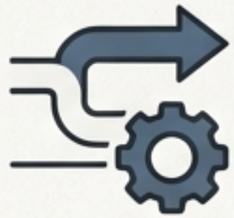
Evaluation on TEST Data

We score the `TEST` partition and compare the model's predictions against the ground-truth `IS_ANOMALY` labels.

	Precision	Recall	F1-Score
0	0.00	0.70	0.90
1	0.00	0.70	0.70
Linear Score	Precision	0.85	0.88
	Standard	0.50	0.50



Performance is measured using a classification report (precision, recall, F1) and ROC AUC, confirming the model effectively identifies the synthetic anomalies.



Scoring PROD Data & Persisting Results

The validated model is applied to the `PROD` data partition. The results, `PRED_IS_ANOMALY` (0/1) and `ANOMALY_SCORE` (a continuous value), are written back to a new table:

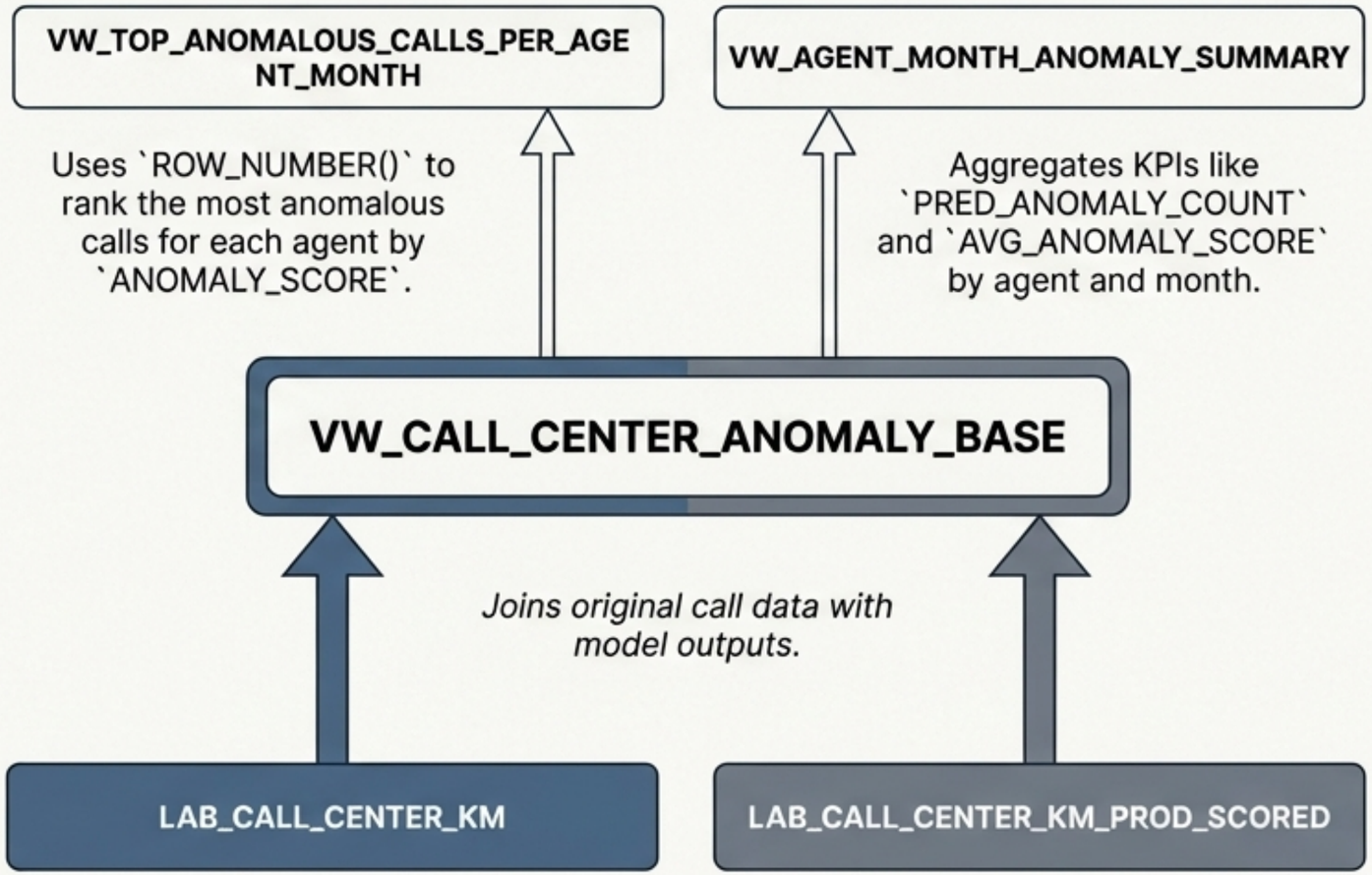
`LAB_CALL_CENTER_KM_PROD_SCORED`

Building a Semantic Layer with SQL Views for Simplified Analytics

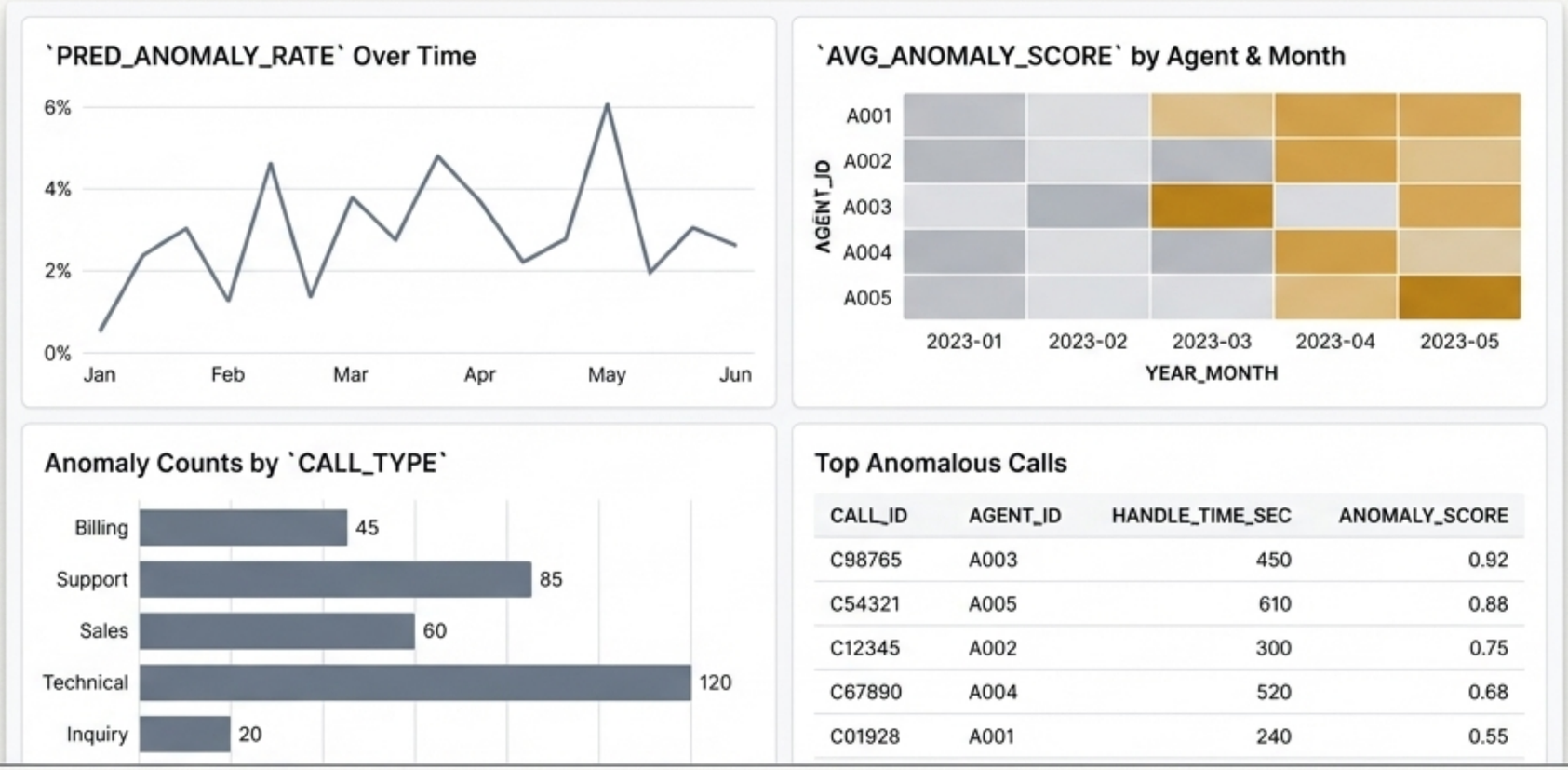
Objective

To provide a clean, unified data source for BI tools and downstream analysis without exposing the complexity of underlying tables.

Source Sans Pro Regular



Visualizing Anomalies: The Detection Dashboard



What we know

We can now precisely identify which agents, call types, and time periods are experiencing the most anomalies.

What we're missing

The quantitative data tells us *what* is anomalous, but not ***why***. This requires manual investigation.

From ‘What’ to ‘Why’: Introducing Snowflake Cortex for AI Explanations

The Game-Changer

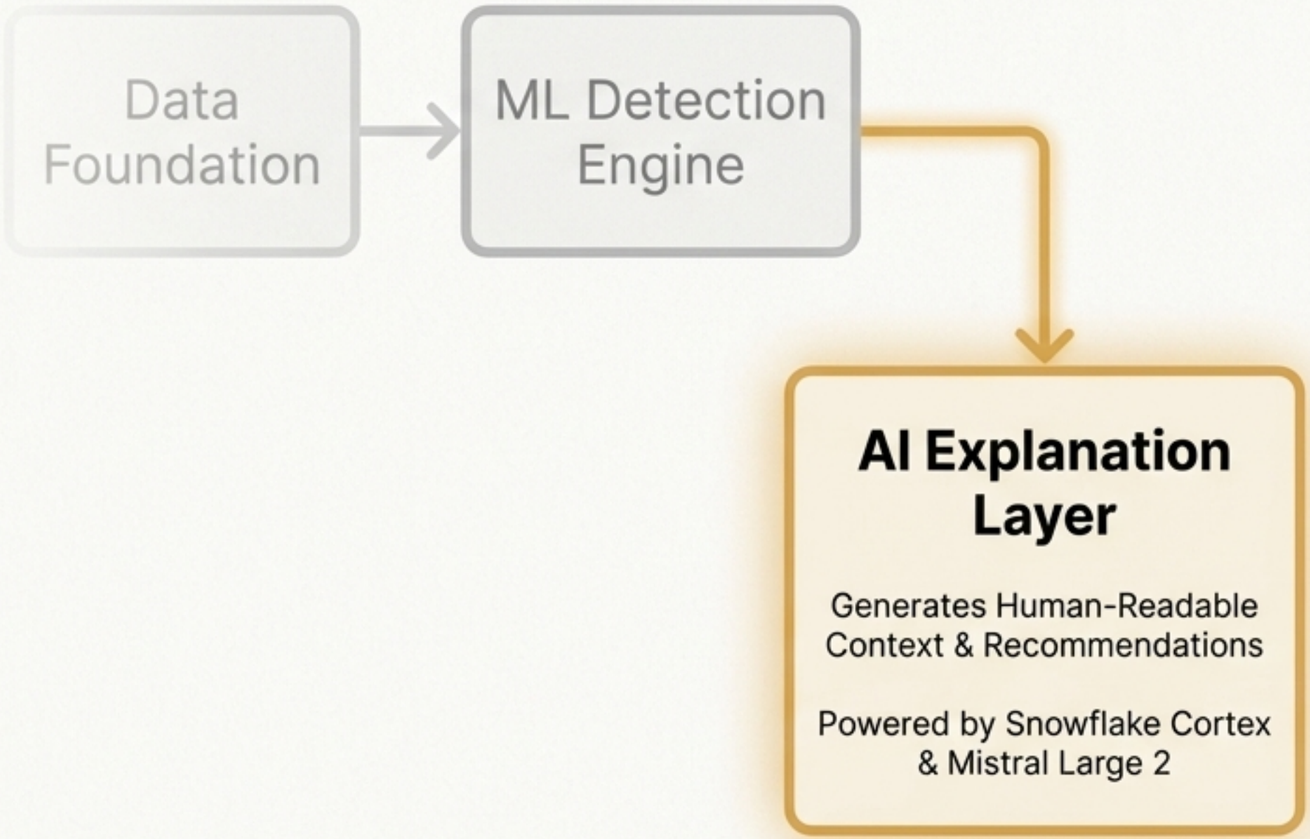
Snowflake Cortex provides built-in LLM capabilities directly via SQL functions, allowing us to generate narrative context without moving data out of the platform.

Core Function

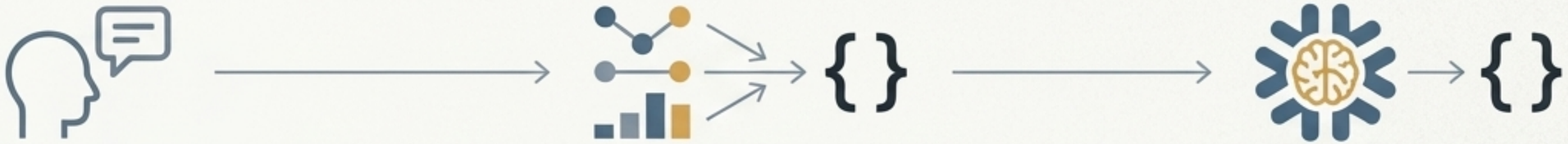
`SNOWFLAKE.CORTEX.COMPLETE` with the `` `mistral-large2` `` model.

Objective

Convert raw metrics and anomaly scores into plain-language, human-readable explanations and recommendations.



Generating Per-Call Explanations with a Cortex Complete Function



Step 1: Define a Persona

The prompt begins by instructing the model: "You are a senior call center performance analyst..."

Step 2: Construct a Payload

For each anomalous call in `VW_CALL_CENTER_ANOMALY_BASE`, we use `OBJECT_CONSTRUCT(...)` to create a JSON object containing key metrics (`HANDLE_TIME_SEC`, `KM_TIME_SEC`, `ANOMALY_SCORE`, etc.).

Step 3: Invoke the LLM

We pass this JSON payload to the `SNOWFLAKE.CORTEX.COMPLETE('mistral-large2', ...)` function.

Expected Output

Call ID: 12345-ABC

This call is anomalous due to an exceptionally high handle time (3200s), which is 5x the agent's average. This suggests a complex customer issue or procedural difficulty.

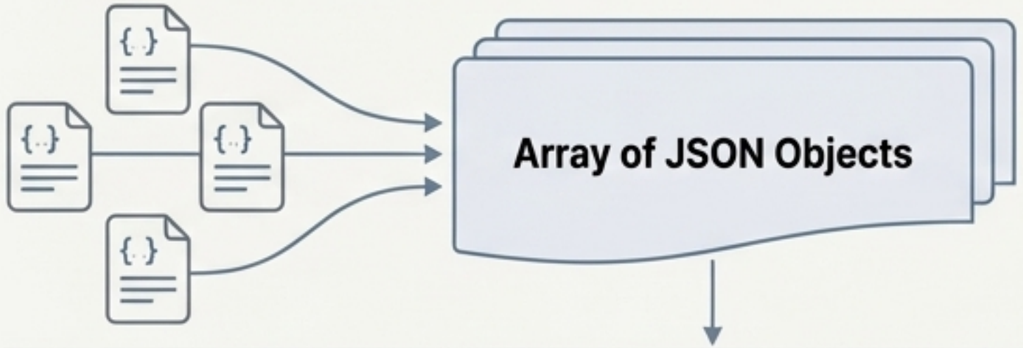
Coaching action: Review call recording with agent to identify knowledge gaps in handling this call type. Assess if KM articles need updating.

Uncovering Systemic Patterns with Aggregate-Level Summaries

Beyond Single Calls: We can use Cortex to analyze trends across multiple data points by feeding it aggregated data.

Example Use Case: Input

Individual agent summaries from 'VW_AGENT_MONTH_ANOMALY_SUMMARY'




```
[
  { "agent_id": 101, "month": "2023-11", "pred_anomaly_rate": 0.25, ... },
  { "agent_id": 102, "month": "2023-11", "pred_anomaly_rate": 0.05, ... },
  { "agent_id": 103, "month": "2023-11", "pred_anomaly_rate": 0.31, ... }
]
```

Cortex Task

Analyze the provided data to identify agents most at risk, describe common patterns across their anomalies, and recommend systemic interventions.

Cortex

Generated Narrative Summary

Key Observation:

Agents 101 and 103 consistently show the highest anomaly rates, primarily driven by long handle times on 'Billing Inquiry' calls.

Common Pattern:

These anomalies frequently occur during the last week of the month, suggesting a potential issue with end-of-month promotions or system performance.

Systemic Recommendation:

Propose a targeted training session for all agents on the 'Billing Inquiry' workflow and investigate the performance of the CRM tool during peak, end-of-month periods.

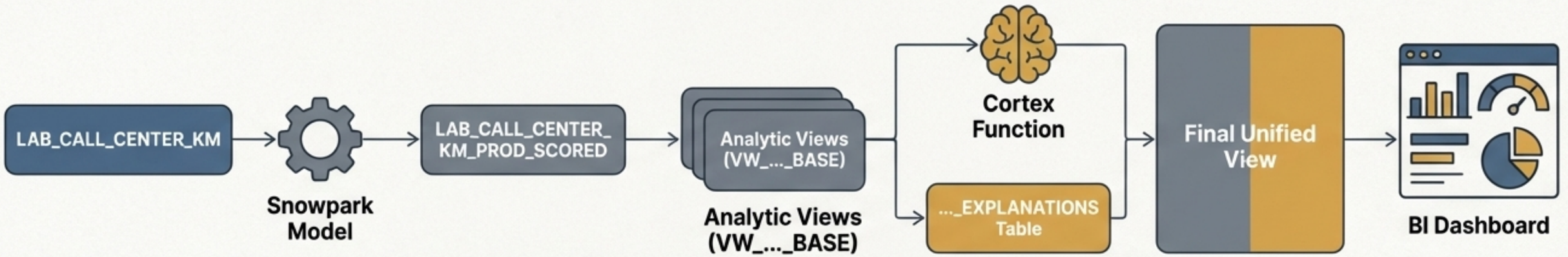
The Complete Solution: Integrating AI Narratives into the BI Layer

VW_CALL_CENTER_ANOMALY_BASE
(scores and metrics)

CALL_CENTER_ANOMALY_EXPLANATIONS
(Cortex-generated text)

VW_CALL_CENTER_ANOMALY_WITH_EXPLANATION
(Unified View)

The End-to-End Snowflake Workflow: From Data to Explained Insight



1. **Data Generation:** Create a labeled synthetic dataset in Snowflake.
2. **Model Training & Scoring:** Use Snowpark to build and apply an anomaly detection model.
3. **Semantic Layer:** Construct SQL views to simplify data for analytics.
4. **BI Visualization:** Identify what is anomalous in dashboards.
5. **AI Explanation:** Use Cortex to generate narratives explaining why.

Value Realized and Future Roadmap

This solution demonstrates a complete, secure, and efficient workflow entirely within Snowflake, transforming raw data into not just detected anomalies, but understood and actionable insights.

Automation

Schedule daily scoring and explanation jobs using Snowflake tasks.

Model Expansion

Incorporate additional anomaly types, such as unusual KM usage patterns or abnormal call escalations.

Operational Integration

Connect the insights directly to workforce management or QA platforms to trigger and track follow-up actions.